SOI: 1.1/TAS DOI: 10.15863/TAS

International Scientific Journal

Theoretical & Applied Science

p-ISSN: 2308-4944 (print) **e-ISSN:** 2409-0085 (online)

Year: 2019 **Issue:** 05 **Volume:** 73

Published: 10.05.2019 **http://T-Science.org**

SECTION 2. Applied mathematics. Mathematical modeling.

QR – Issue QR – Article

Meruyet Murat
master student,
Taraz State University named after M.Kh. Dulati

Alexandr Shevtsov
candidate of technical sciences, member of PILA (USA),
department of «Mathematics», deputy director on Science
of faculty of Information technologies,
automation and telecommunications,
Taraz State University named after M.Kh. Dulati
Shev_AlexXXXX@mail.ru

LIBRARY FOR SOME NUMERICAL OPTIMIZATION ALGORITHMS

Abstract: *In the article the library for some algorithms of realization of optimization problems in the system of computer algebra Maple is developed.*

Key words: *Maple, equation, library.*

Language: *English*

Citation: Murat, M., & Shevtsov, A. (2019). Library for some numerical optimization algorithms. *ISJ Theoretical & Applied Science, 05 (73),* 254-265.

Soi: http://s-o-i.org/1.1/TAS-05-73-36 ***Doi***: crossref https://dx.doi.org/10.15863/TAS.2019.05.73.36

Introduction

Some numerical methods for solving the optimization problem are based on the exact or approximate calculation of its characteristics cannot be solved by the standard library Maple [1-10]. Therefore, by collecting separate algorithms for solving the optimization problem in a separate library, we can contribute to the solution of this problem [9-11].

Let us consider the process of developing a library to implement an optimization algorithm in a Maple computer algebra system.

Materials and Methods

The development of a library of Maple NumOpt_MuratM.MapleLib for solving some numerical optimization problems

Library development NumOpt_MuratM. MapleLib implementing the chord method

Numopt library code:

```
> restart:
  NumOpt := table():
  NumOpt[horda] :=proc(f, otrezok, eps, M)
   local x, a, b, c, ff, i:
  x := lhs(otrezok):
  a := op( rhs( otrezok ) )[1]:
  b := op( rhs( otrezok ) )[2]:
  ff := unapply(evalf(f), x):
  c := b - ff(b)*(b-a)/(ff(b)-ff(a)):
  if (ff(a)*ff(b) > 0)
  then print("неверно введённый интервал")
   else
  if (ff(a)*ff(b) < 0)
  then
  while (abs(ff(c)/M) > eps and f(c) ≠ 0)
   do
  if (ff(a)*ff(c) > 0)
  then a := c:
  else
  b := c:
  fi:
  c := b-ff(b)*(b-a)/(ff(b)-ff(a)):
  end do:
  else
  if (ff(a)=0)
  then c := a:
  else
   c := b:
    end if:
    end if:
  x := evalf(c):
  end if:
   end proc:
```

Connect the library

```
> read(`D:/NumOpt_MuratM.MapleLib`);
  with(NumOpt);
```

[bisection, dixotomi, gold, horda, newton, newton_mod]

Code using the library NumOpt for method chords:

$$f := x \to x - 5\sin\left(\frac{1}{(x-1)^2+1}\right); fl := x - 5\sin\left(\frac{1}{(x-1)^2+1}\right);$$

$$a := 0; b := 3.5; eps := 10^{-3};$$

$$f := x \mapsto x - 5\sin\left(\frac{1}{(x-1)^2+1}\right)$$

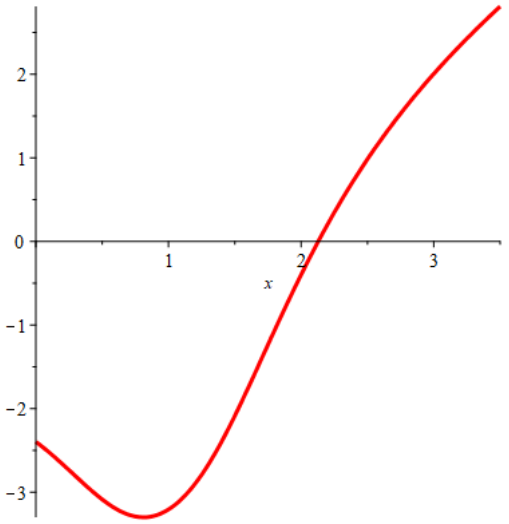
$$fl := x - 5\sin\left(\frac{1}{(x-1)^2+1}\right)$$

$$a := 0$$

$$b := 3.5$$

$$eps := \frac{1}{1000}$$

$$graph := plot(fl, x = a..b, color = red, thickness = 3);$$

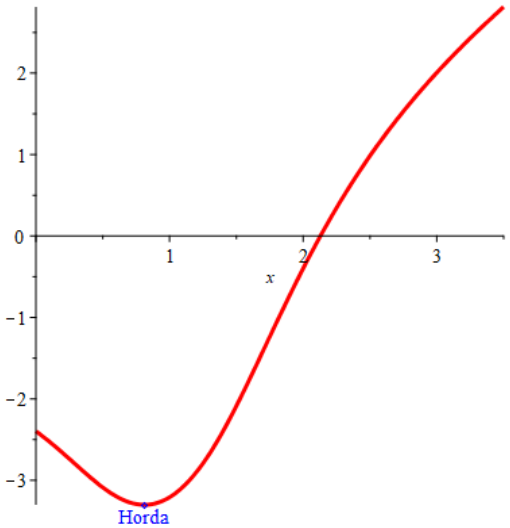



```
f2 := diff(f1, x);
x_xopda := horda(f2, x = a..b, eps, 1);
y_xopda := f(x_xopda);
with(plots) :
p := pointplot([x_xopda, y_xopda], color = blue) :
p1 := textplot([x_xopda, y_xopda,'typeset'("Horda")],'align'='below', color = blue) :
display(graph, p1, p);
```

$$f2 := 1 + \frac{5\,(2x-2)\cos\left(\dfrac{1}{(x-1)^2+1}\right)}{\left((x-1)^2+1\right)^2}$$

$$x\_xopda := 0.8113132771$$

$$y\_xopda := -3.300699391$$



Library development NumOpt_MuratM.MapleLib implementing the Golden section method

Numopt library code:

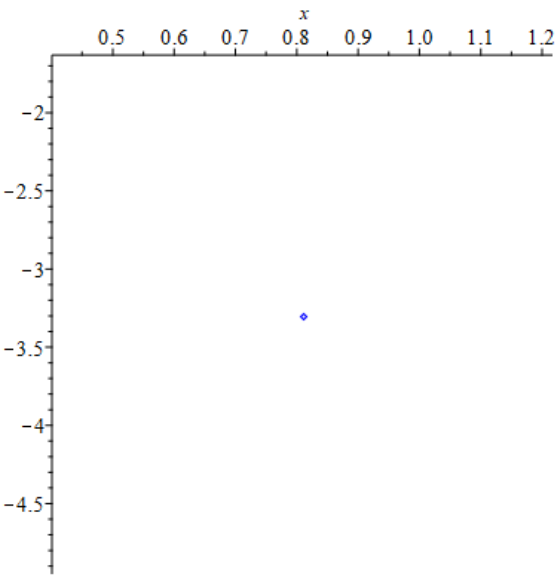
```
> NumOpt[gold]:=proc(a, b, eps::numeric,
  f::procedure, niter::evaln)
  local n, aa, bb, ab, xa, xb, fa, fb, fxa, fxb, r;
  global DEB;
  r := evalf(sqrt(5)-1)/2;
  fa := f(a); fb := f(b);
  aa := a; bb := b;
  xa := bb-r*(bb-aa); xb := r*(bb-aa)+aa;
  fxa := f(xa); fxb := f(xb);
  for n from 1 to 100
  while (abs(bb-aa)>eps) do
  if (fxb<fxa) then
  aa := xa; fa := fxa;
  xa := xb; fxa := fxb;
  xb := bb+aa-xa;
  fxb := f(xb);
  else
  bb := xb; fb := fxb;
  xb := xa; fxb := fxa;
  xa := bb+aa-xb;
  fxa := f(xa);
  fi;
  ab := (aa+bb)/2;
  if DEB=1 then print(n, ab, f(ab)); fi;
  od;
  niter := n-1;
  ab;
   end:
```

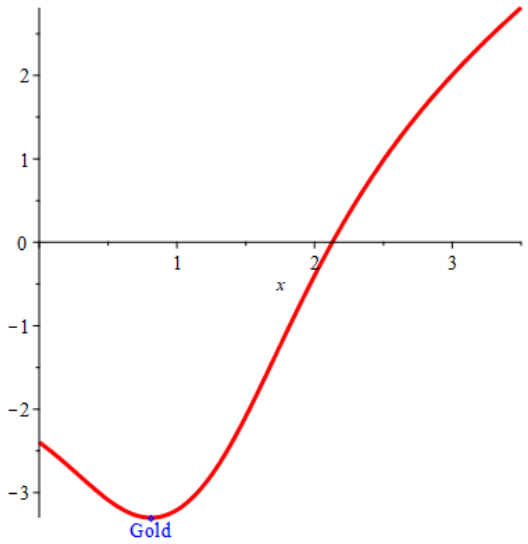
Code using the library NumOpt for the method of the Golden section:

```
x_золотоесечение := gold(evalf(a), evalf(b), eps, f, n);
y_золотоесечение := f(x_золотоесечение);
with(plots):
s := pointplot([x_золотоесечение, y_золотоесечение], color = blue);
s1 := textplot([x_золотоесечение, y_золотоесечение, typeset("Gold")], align='below', color = blue):
display(graph, s, s1);
```

$$x\_золотоесечение := 0.8113338800$$

$$y\_золотоесечение := -3.300699407$$





Library development NumOpt_MuratM.MapleLib performing the method of dichotomy

Numopt library code:

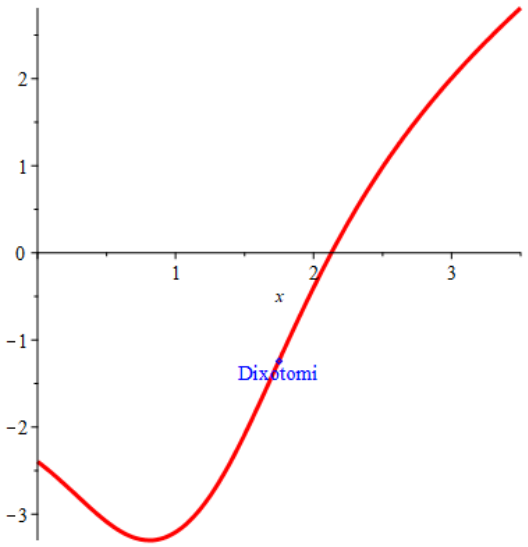
```
> NumOpt[dixotomi]:=proc(f, otrezok, epsilon)
   local x, a, b, c, ff, i:
  x := lhs(otrezok):
   a := op( rhs( otrezok ) )[1]:
  b := op( rhs( otrezok ) )[2]:
  ff := unapply(evalf(f), x):
  if (ff(a)*ff(b)>0)
  then print("неверно введённый интервал"):
   else
    if (ff(a)*ff(b)<0)
    then
  while (abs(b-a)>epsilon and f(c)<>0)
     do
  c := (a+b)/2:
     if (ff(a)*ff(b)>0)
     then a := c:
     else
     b := c:
     fi:
  end do:
  else
   if (ff(a)=0)
  then c := a:
       else
            c := b:
         end if:
    end if:
  x := evalf(c);
  end if:
   end proc:
```

Code using the library NumOpt for the method of dichotomy:

```
x_дихотоми := dixotomi(f1, x = a..b, eps);
y_дихотоми := f(x_дихотоми);
with(plots) : e := pointplot([x_дихотоми, y_дихотоми], color = blue) :
e1 := textplot([x_дихотоми, y_дихотоми,'typeset'("Dixotomi")],'align'='below', color = blue) :
display(graph, e, e1);
```

$$x\_дихотоми := 1.749145508$$

$$y\_дихотоми := -1.238937316$$



Library development NumOpt_MuratM.MapleLib implementing Newton's method

Numopt library code

```
> NumOpt[newton]:=proc(x0, eps, nmax)
  local x, dx, ff, i;
  x := x0;
  ff := f(x);
  dx := 2*eps;
  i := 0;
  #print (i,x,dx,ff);
  printf(`i=%2d x[%2d]=%8.5f dx=%6.0e f(%8.5f)=%10.5f
  \n`, i, i, x, dx, x, ff);
  for i to nmax while abs(dx)>eps do
  dx := evalf(ff/df_f(x));
  x := x-dx;
  ff := f(x);
  #print (i,x,dx);
  printf(`i=%2d x[%2d]=%8.5f dx=%6.0e f(%8.5f)=%10.5f \n`, i, i, x,
     dx, x, ff);
  od;
  printf(`Answer x[%2d]=%8.5f`, i-1, x);
  return x;
   end proc:
```

Code for using the NumOpt library for the Newton method

```
fa := f(a);
df := diff(f1, x$1);
subs(x = a, df);
df_f := unapply(df, x);
 evalf(df_f(a));
x_ньютон := newton(0.5, eps, n);
y_ньютон := f(x_ньютон);
with(plots) : r := pointplot([x_ньютон, y_ньютон], color = blue) :
r1 := textplot([x_ньютон, y_ньютон,'typeset'("Newton")],'align'='below', color = blue) :
display(graph, r, r1);
```

$$fa := -5\sin\left(\frac{1}{2}\right)$$

$$df := 1 + \frac{5\,(2x-2)\cos\left(\dfrac{1}{(x-1)^2+1}\right)}{\left((x-1)^2+1\right)^2}$$

$$1 - \frac{5\cos\left(\frac{1}{2}\right)}{2}$$

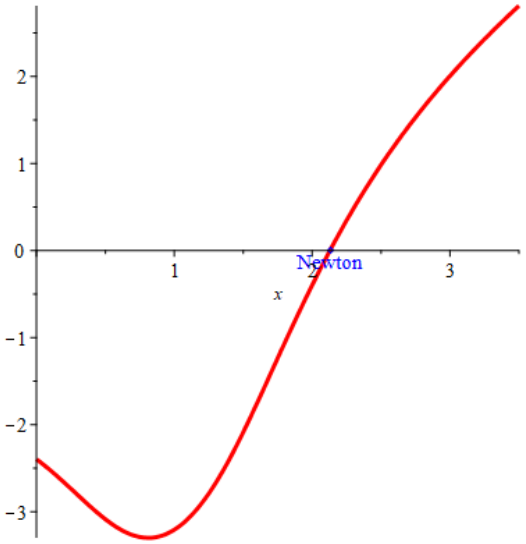
$$df\_f := x \mapsto 1 + \frac{5\,(2x-2)\cos\left(\dfrac{1}{(x-1)^2+1}\right)}{\left((x-1)^2+1\right)^2}$$

$$-1.193956405$$

```
                                  -1.193956405
i= 0 x[ 0]= 0.50000 dx= 2e-03 f( 0.50000)=  -3.08678
\ni= 1 x[ 1]=-2.01068 dx= 3e+00 f(-2.01068)=  -2.50667 \ni= 2 x[
2]= 1.54879 dx=-4e+00 f( 1.54879)=  -1.92662 \ni= 3 x[ 3]= 2.12730
dx=-6e-01 f( 2.12730)=  -0.00409 \ni= 4 x[ 4]= 2.12867 dx=-1e-03 f
( 2.12867)=  -0.00000 \ni= 5 x[ 5]= 2.12867 dx=-6e-07 f( 2.12867)=
 -0.00000 \nAnswer x[ 5]= 2.12867
```

$$x\_ньютон := 2.128671429$$

$$y\_ньютон := -1.10^{-9}$$



Library development NumOpt_MuratM.MapleLib implementing the modified Newton method

Numopt library code:

```
> NumOpt[newton_mod]:=proc(x0, eps, nmax)
  local x, dx, ff, df0, i;
  x := x0;
  ff := f(x);
  df0 := df_f(x);
  dx := 2*eps;
  i := 0;
  #print (i,x,dx,ff);
  printf(`i=%2d x[%2d]=%8.5f dx=%6.0e f(%8.5f)=%10.5f \n`, i, i, x,
     dx, x, ff);
  for i to nmax while abs(dx) > eps do
  dx := evalf(ff/df0);
  x := x-dx;
  ff := f(x);
  #print (i,x,dx);
  printf(`i=%2d x[%2d]=%8.5f dx=%6.0e f(%8.5f)=%10.5f \n`, i, i, x,
     dx, x, ff);
  od;
  printf(`Answer x[%2d]=%8.5f`, i-1, x);
  return x;
   end proc:
```

Library development NumOpt_MuratM.MapleLib implementing the bisection method

Numopt library code:

```
> NumOpt[bisection]:=proc(Eq, otrezok , eps)
  local f, a, b, i, c:
  Digits := length(floor(a))-log[10](eps) + 2 :
  f := unapply(Eq, x) :
   a := op( rhs( otrezok ) )[1]:
  b := op( rhs( otrezok ) )[2]:
   for i from 0 while (b − a) > eps do
   c := ((a + b) / 2) :
  if evalf(f(a) * f(c)) < 0 then b := c :
  else a := c :
  fi:
  od:
   end proc:
```

Next, save the created library to the computer disk:

```
save(NumOpt, `D:/ NumOpt_MuratM.MapleLib`);
```

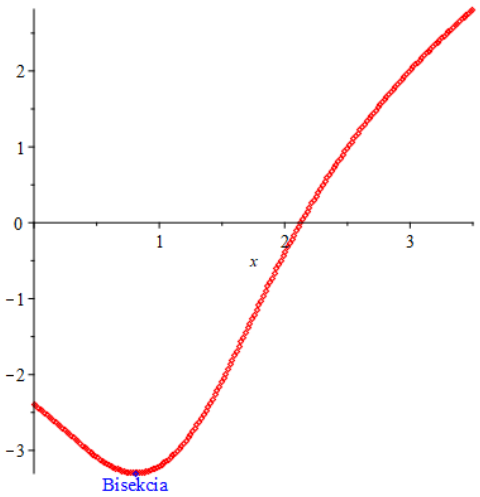
Code for using the NumOpt library for the bisection method:

```
Eq := diff(f1, x);
x_бисекция := bisection(Eq, x = evalf(a) ..evalf(b), eps);
y_бисекция := f(x_бисекция);
with(plots) :
m := pointplot( [x_бисекция, y_бисекция], color = blue) :
m1 := textplot( [x_бисекция, y_бисекция,'typeset'("Bisekcia") ],'align'='below', color = blue) :
 display(graph, m, m1);
```

$$Eq := 1 + \frac{5\,(2x-2)\cos\left(\frac{1}{(x-1)^2+1}\right)}{\left((x-1)^2+1\right)^2}$$

$$x\_бисекция := 0.8109130859375000$$

$$y\_бисекция := -3.300698618$$



Library development NumOpt_MuratM.MapleLib implementing the search for a local minimum by the method of half division

```
NumOpt[LocalMinimum]:=proc(f, otrezok , eps)
 local f, a, b, i, c:
Исходные функции
f:= 12*surd(6*(x-1)^2, 3)/((x+1)^2+8):
f1:= 12*surd(6*(cl[i]-1)^2, 3)/((cl[i]+1)^2+8):
f2:= 12*surd(6*(cr[i]-1)^2, 3)/((cr[i]+1)^2+8):
Отрезки локализации
a1:=-1: b1:= 2:
Точность шагов
prec:= 0.001:
Задание левой и правой начальной точки, а также области
i:= 1:
l[i]:= a1:
r[i]:= b1:
Area[i]:= r[i]-l[i]:
while Area[i]>2.4*prec do
cl[i]:= (l[i] + r[i])/2 - prec:
cr[i]:= (l[i] + r[i])/2 + prec:
Определение новых точек
l[i+1]:= `if`(f1 > f2, cl[i], l[i]):
Определяем новую область
r[i+1]:= `if`(f1 ≤ f2, cr[i], r[i]):
Определяем новую область
Area[i+1]:= r[i+1]-l[i+1]:
Определяем x как возможное решение
x:= (r[i+1]+l[i+1])/2:
rezx1:= x:
rezy1:= f:
i:= i+1;
 end do:
  "Первый локальный минимум:"; "Абсцисса:";
 rezx1;
 "Ордината:";
  rezy1;
 end proc:
```

Library development NumOpt_MuratM.MapleLib implements search of local maxima and global maximum by the method of division in half

```
 "Абсцисса:";
 rezx1;
 "Ордината:";
 rezy1;
 i:= 1:
 l[i]:= a2:
 r[i]:= b2:
 Area[i]:= r[i]-l[i]:
 Вычисление второго максимума
 while Area[i]>2.4*prec do
 cl[i]:= (l[i] + r[i])/2 - prec:
 cr[i]:= (l[i] + r[i])/2 + prec:
 l[i+1]:= `if`(f1 < f2, cl[i], l[i]):
 r[i+1]:= `if`(f1 ≥ f2, cr[i], r[i]):
 Area[i+1]:= r[i+1]-l[i+1]:
 x:= (r[i+1]+l[i+1])/2:
 rezx2:= x:
 rezy2:= f:
 i:= i+1;
 end do:
 "Второй локальный максимум";
 "Абсцисса:";
 rezx2;
 "Ордината:";
 rezy2;
 "Глобальный максимум";
 "Абсцисса:";
 xgl:= `if`(rezy1>rezy2, rezx1, rezx2): xgl;
 "Ордината:";
```

```
NumOpt[LocalGlobalMaximum]:=proc(f, otrezok , eps)
local f, a, b, i, c:
 Исходные функции
f:= 12*surd(6*(x-1)^2, 3)/((x+1)^2+8):
f1:= 12*surd(6*(cl[i]-1)^2, 3)/((cl[i]+1)^2+8):
f2:= 12*surd(6*(cr[i]-1)^2, 3)/((cr[i]+1)^2+8):
Отрезки локализации
a1:=-3: b1:= 0:
a2:= 1: b2:= 4:
Точность шагов
prec:= 0.001:
i:= 1:
l[i]:= a1:
r[i]:= b1:
Area[i]:= r[i]-l[i]:
while Area[i]>2.4*prec do
cl[i]:= (l[i] + r[i])/2 - prec:
cr[i]:= (l[i] + r[i])/2 + prec:
l[i+1]:= `if`(f1 < f2, cl[i], l[i]):
r[i+1]:= `if`(f1 ≥ f2, cr[i], r[i]):
Area[i+1]:= r[i+1]-l[i+1]:
x:= (r[i+1]+l[i+1])/2:
rezx1:= x:
rezy1:= f:
i:= i+1;
end do:
"Первый локальный максимум:";
```

```
ygl:= `if`(rezy1>rezy2, rezy1, rezy2):ygl;
 end proc:
```

Library development for the minimum function of two variables "by the NumOpt_MuratM.MapleLib implements the search method of the fastest descent»

```
NumOpt[SpuskLocalMinimum]:=proc(f, otrezok , eps)
local f, a, b, i, c:
f:= 1.2*x1^2 + 1.8 * x2^2 - 4 * x1 - 4 * x2:
Определяем производные по двум переменным
df1:= diff(f, x1);
df2:= diff(f, x2);
x1prec:= solve(df1, x1):
x2prec:= solve(df2, x2):
Начальное приближение
X1[1]:=-1:
X2[1]:=-1:
prec:= 0.01:
i:= 1:
x1rez:= 0:
x2rez:= 0:
while x1rez=0 do
x1:= X1[i]:
x2:= X2[i]:
x1rez:= `if`(df1<prec,(`if`(df1>-prec, X1[i], 0)), 0):
x2rez:= `if`(df2<prec,(`if`(df2>-prec, X2[i], 0)), 0):
X1[i+1]:= `if`(df1>0, X1[i]-prec, X1[i]+prec):
X2[i+1]:= `if`(df2>0, X2[i]-prec, X2[i]+prec):
i:= i+1:
end do:
```

```
"Точные значения:";
x1 := x1prec;
x2 := x2prec;
"Точка минимума:";
x1 := x1rez;
x2 := x2rez;
"Значение функции в минимуме:";
f;
  end proc:
```

The considered methods give high accuracy of finding extremums of the function.

	Значения, найденные средствами Maple 7	Найденные методом половинного деления	Разница в результатах Вычислений
min	(1 , 0)	(1.000093628 , 0.003746629028)	0.000093628 , 0.003746629028
1 max	(-2.0 , 5.039684200)	(-2.000093628 , 5.039684194)	0.000093628 , 0.000000006
2 max	[{ x = 3.0 }, 1.442249570]	(3.000093628 , 1.442249569)	0.000093628 , 0.000000001

Значения, найденные методом наискорейшего спуска	Значения, найденные средствами Maple 7
x1 := 1.67 x2 := 1.11 y= -5,55554	[{ x1 = 1.666666667 , x2 = 1.111111111 }, −5.555555555]

Conclusion

As a result of the study, the algorithms of numerical methods for solving equations and solving numerical optimization problems based on the methods of half division, search for extrema functions, as well as the method of the steepest descent were studied. It can be concluded that:

• Algorithms on Maple for solving numerical optimization problems are obtained

• Numopt_muratm Library was developed.MapleLib implements numerical methods for solving equations and numerical optimization

• The library has been tested on several examples

• The resulting library can be used in research and calculations.

References:

1. Serikbaev, N., & Shevtsov A. (2018). The development of a library of delphi for the solution of transcendental equations. *ISJ Theoretical & Applied Science, USA, 05 (61)*, 1-10. Doi: https://dx.doi.org/10.15863/TAS.2018.05.61.1
2. Zhunisbekov, S., Shevtsov, A., & Karymsakov, N. (2018). Development of the simplest perceptron for artificial intelligence. *ISJ Theoretical & Applied Science, 06 (62)*, 1-10. Soi: http://s-o-i.org/1.1/TAS-06-62-1 Doi: https://dx.doi.org/10.15863/TAS.2018.06.62.1
3. Kudryavtseva, I. V., Rykov, S., Rykov, S. V., & Skobov, E. D. (2014). *Optimization methods in the examples in the MathCAD 15 package.* Part I:Studies'. manual. (p.166). SPb.: NRU ITMO, Aibt.
4. Shevtsov, A. N., et al. (2017). Innovative machines for post-harvest grain handling. *ISJ Theoretical & Applied Science, 01 (45)*, 171-175. Soi: http://s-o-i.org/1.1/TAS-01-45-32

Doi: https://dx.doi.org/10.15863/TAS.2017.01.45.32

5. Shevtsov, A., & Talaybekkyzy, N. (2017). Development of autonomous robot for radiation reconnaissance. *ISJ Theoretical & Applied Science, 03 (47)*, 1-4. Soi: http://s-o-i.org/1.1/TAS-03-47-1 Doi: https://dx.doi.org/10.15863/TAS.2017.03.47.1
6. Abenov, U., & Shevtsov, A. (2017). The application of system maple and delphi to solve problems of discrete mathematics. *ISJ Theoretical & Applied Science, 05 (49)*, 101-119. Soi: http://s-o-i.org/1.1/TAS-05-49-17 Doi: https://dx.doi.org/10.15863/TAS.2017.05.49.17
7. Abenov, U., & Shevtsov, A. (2017). The practical aspect of using greedy algorithm. *ISJ Theoretical & Applied Science, 05 (49)*, 120-127. Soi: http://s-o-i.org/1.1/TAS-05-49-18 Doi: https://dx.doi.org/10.15863/TAS.2017.05.49.18
8. Shevtsov, A., et al. (2017). Normalization of freshly harvested grain. *ISJ Theoretical & Applied Science, 05 (49)*, 249-256. Soi: http://s-o-i.org/1.1/TAS-05-49-40 Doi: https://dx.doi.org/10.15863/TAS.2017.05.49.40
9. Shevtsov, A. (2017). Development of an automatic dust collection system in mines. *ISJ Theoretical & Applied Science, 08 (52):* 1-4. Soi: http://s-o-i.org/1.1/TAS-08-52-1 Doi: https://dx.doi.org/10.15863/TAS.2017.08.52.1
10. Zhunisbekov, S., Shevtsov, A., & Kairliyeva, D. (2017). On some aspects of the implementation of the principle of hamilton in maple. *ISJ Theoretical & Applied Science, 09 (53)*, 89-92. Soi: http://s-o-i.org/1.1/TAS-09-53-15 Doi: https://dx.doi.org/10.15863/TAS.2017.09.53.15
11. Koybakov, S., Maliktayuly, M., & Shevtsov, A. (2017). Simulation models of chaotic flow. *ISJ Theoretical & Applied Science, 12 (56)*, 87-92. Soi: http://s-o-i.org/1.1/TAS-12-56-16 Doi: https://dx.doi.org/10.15863/TAS.2017.12.56.1
12. Serikbaev, N., & Shevtsov, A. (2017) the development of a library of maple for the solution of transcendental equations. *ISJ Theoretical & Applied Science, 12 (56)*, 261-279. Soi: http://s-o-i.org/1.1/TAS-12-56-42 Doi: https://dx.doi.org/10.15863/TAS.2017.12.56.42
13. Murat, M., & Shevtsov, A. (2018). On some numerical optimization algorithms. ISJ Theoretical & Applied Science, 12 (68), 1-5. Soi: http://s-o-i.org/1.1/TAS-12-68-1 Doi: https://dx.doi.org/10.15863/TAS.2018.12.68.1